Program Workshop

2008년 1월 2일 수요일

오후 2:23

Contents

Introduction, Background, System Overview

2008년 1월 8일 화요일

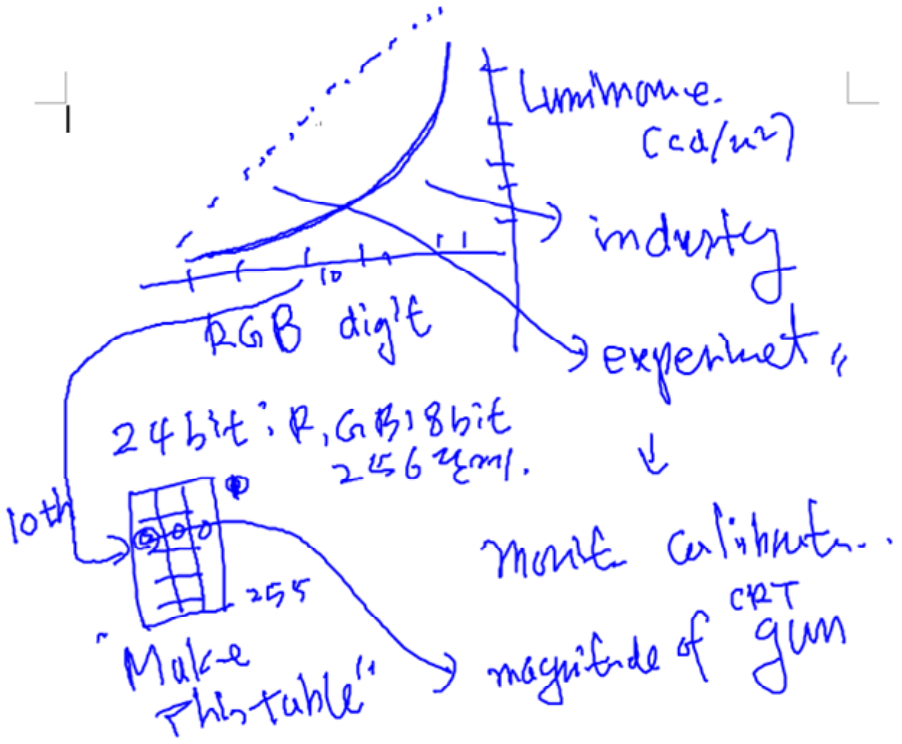
오후 7:00

1. Introduction
   a. Date and Time : Tue, Thu, (Sat) 2 hours in Jan 2008 (6 times)
   b. Instructor: Hyunggoo Kim
   c. Assumption
      i. Novice to Matlab and experimental programming
      ii. General knowledge about compuer system and programming
   d. Objective
      i. Being able to debug experimental software and troubleshoot H/W, S/W problems
   e. Attendees
      i. Read related documents
      ii. Be accustomed to termonogy through googling
   f. Contributors
      i. Please send email to me if you want to contribute to the development.
   g. Related documents
      i. Brief program features and appearance: "KM archive description" http://147.46.14.210/~k09/snags/en/KM_intro.pdf
2. Programming and Toolbox fundamentals - 2008/01/08 Tue
   a. Matlab basics
      i. Script language / easy graphics
         1) Funcions are compiled before execution and loaded to memory
         2) Easy debugging with integrated editor
            a) Use 'breakpoint' and 'step', see variables
      ii. Extensive Toolbox
         1) Lots of numerical analysis functions
      iii. References
         1) Mastering Matlab
         2) "Doc" command - extensive explanation about all math to implementation Composed of three devisions
            a) Matlab languages
            b) Toolboxes
            c) Demos
   b. Data acqusition http://zone.ni.com/devzone/cda/tut/p/id/3216
      i. H/W interface for signal flow
         1) External signal source -> BNC cable
         2) BNC cable -> National Instrument board
         3) NI board -> National Instrument PCI card -> Analog to digital conversion
         4) H/W installation can be checked with 'Device Manager (장치관리자)'
      ii. S/W interface
         1) NI-DAQ device driver reads digital data in PCI card to memory
         2) Matlab Data Acquisition Toolbox reads data using device driver
         3) Normal operation of NI-DAQ device driver can be verified with NI Measurement & Automation Explorer
      iii. Current NI board, PCI card specs: see "Current Platform Specification"
      iv. DAQ channel – multiplexer, terminater: double data size
         1) Photodiode signal is significantly distorted by the previous channel
         2) Crosstalk http://zone.ni.com/devzone/cda/tut/p/id/630
         3) See also "Citation in channel coupling problem"
      v. Terminology
         1) Sampling rate, signal error, nyquist theorem
         2) Differential, single ended mode
   c. GUI

i. For efficient manipulation of experiment and analysis, GUI are often used.
ii. Two ways to create experimental GUIs
    1) Create by function (i.e., uicontrol)
    2) Using Guide framework. The latter is much easier and convenient for debugging
iii. Guide: Matlab GUI interface toolkit
iv. Event-driven programming
    1) Execute callback function when event occurs (click button, mouse on, etc)
    2) Enables multi-thread programming
        a) Two functions can be executed simultaneously
        b) Used in RTMonitor network mode
v. http://www.mathworks.com/access/helpdesk/help/techdoc/index.html
   See "Creating Graphical User Interfaces" chapter

d. Stimulation http://psychtoolbox.org/PTB-2/intro.html#cite
i. Psychphysics toolbox: Matlab toolbox for direct display control
ii. Frame-level stimulus display, Color lookup table, Keyboard input
iii. Currently uses version 2.54 (upgarde to v3.x may cause compatibility problems)
iv. Terminology
    1) frame buffer, frame rate, CRT interval
    2) Color lookup table, monitor calibration, photodiode
v. References
    1) Brainard, D. H. (1997) The Psychophysics Toolbox, Spatial Vision 10:433-436.
    2) Pelli, D. G. (1997) The VideoToolbox software for visual psychophysics: Transforming numbers into movies, Spatial Vision 10:437-442
    3) And other documents from the above authors

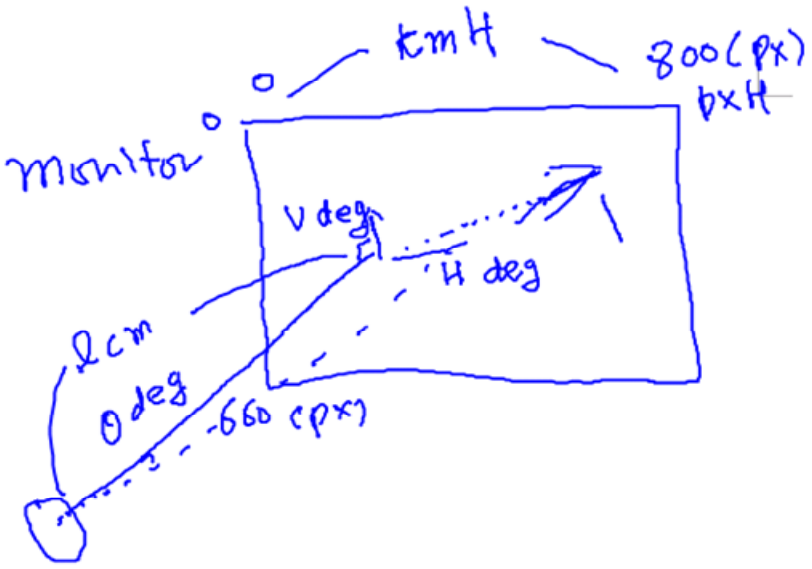
e. Monitor calibration



i. The objective of monitor calibration is to accurately control physical luminance in linear scale
ii. To do this, at first we get empirical Luminance value with respect to given RGB points using chromometer (better if photometer)
iii. Two methods are available
    - Adapt inverse of above function by interpolating above points, and get fixed RGB values
        - Lum2RGB() function RGB2Lum() operates like this (conversion by software)
    - Get inverse RGB value of each 0-255 virtual linear arrays, and get RGB signal

magnitude by indexing the array
    - Color lookup table in the graphic card operates like this. (automatically done by hardware)
iv. Two methods have same precision if hardware color lookup table uses same size (currently, 24-bit) indexing array
v. In our windows machine, psychphysics toolbox 'SetCult', 'SetGamma' function does not operate properly. Maybe a problem of windows psychphysics toolbox, or our graphic card. Therefore, we should it by software method.
vi. 701_stimulator color lookup table structure
    - [RGB_value R_lum G_lum B_lum White_lum; ...]

f. Stimulus position calculation



i. pxDistance = l * pxH / cmH
    - Spatial interval Deg2px() = pxDistance * tand(theta)
ii. Absolute position gaze2px() = [gazefoot(1) + deg2px(theta_h) gazefoot(2) - deg2px(theta_v)
iii. We manipulate monitor H/W view area configuration so that the below satisfies pxH/cmH == pxV / cmV
iv. TODO adapt eccentricity in deg2px() to be more accurate in the peripheral area

g. Communication
i. There are several methods to communicate between two computers - serial port, parallel port, usb port, TCP/IP network, etc.
ii. TCP/IP is most complex communication method.
    1) Cons: it spends more time. Therefore, We should use it not for the synchronization, but for the loose notification.
    2) Pros: we can communicate with string, file regardless of file size
iii. Tcp/udp toolbox - Matlab toolbox for basic character message and file transfer
    1) http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=345&objectType=File

h. Synchronization
i. Synchronization - temporal alignment in different events
ii. Topics about synchronization
    1) External input really occurs at the measured time ?
        a) TCP/IP, serial, parallel, keyboard input
    2) External output really send signals at the right time ?
    3) Can the two machine have synchronized timer ?
    4) NI provides RTSI bus. See "Advanced Synchronization Techniques for Data Acquisition" http://zone.ni.com/devzone/cda/tut/p/id/2997

iii. Strict synchronization can be accomplished by photodiode

i. Performance Tuning for real-time display control
   i. SetPriority function in psychophysics toolbox
   ii. Routines
      1) While - GetSecs : may display 1 more frames (incorrect)
      2) WaitForBlanking, (multiple frames)
         a) Most accurate
         b) But, can do no job while display
      3) While and WaitForBlnaking

```
Count = 2
While
        WaitForBlanking, 1
        Do something ... ( < 5ms)
        Count = Count - 1
        If Count < 0 break
End
----------------------------
```

Caution: If "Do something" take less than 1ms, WaitForBlanking does not wait for next frame, but return immediately for redundant frame. Therefore, There should be some "Time consuming" job if "Do something meaningful" does not executed. WaitSecs() function can be a candidate.

      4) (hard) http://147.46.14.210/~k09/snags/pwiki/index.php?RealTimeExp

3. Overview of the System
   a. Current architecture - diagram
      - Two computer is connected with network hub. IP of the two computer should be in the same subnet (X.X.X.N : N should be same for safe operation in case of outer network failure)
   b. Programs in each machine

Machine	Stimulator		DAQer
Qualitative RF	RotDim	TCP/IP	RTMonitor
Quantitative RF	RFViewer	=> Command =>	RFAnalyzer
Task execution	CalibMainPannel		Trial Iterator

   ○ RFAnalyzer, Trial Iterator is a function module called from RTMonitor
   ○ Stimulator send run mode by string command. RTMonitor interprete command and call module function

   c. Source tree

      - Source tree has hierarchical structure with subdirectory
      - Especially, library functions in bolds are frequently used.
      - Use may find library test routines in the same direcoty named t_<function name>.m or x_<function name>.m. It can be useful to understand the function.

RFMapping/	Receptive field mapping - RFViewer, RFStimulator - RFManager, RFAnalyzer - RF_RotDim
RTMonitor/	Real-time extracellular activity monitor
benchmark/	Performance test of DAQ, display, synchronization
calib/	CalibMainPannel

export/	Exported programs (modified by k09) Hyojung's sorter, Joonyeol's sorter
gallery/	Scripts making stimulus
lib/	Common library (has subdirectories)
+ daq/	DAQ Toolbox wrapper functions
+ display/	Psychphysics toolbox wrapper functions
+ exp_archive/	N/A
+ exp_design/	Experiment design functions
+ eye/	Eye monitor functions
+ net/	Network wrapper functions
+ null_daq/	Substitutional functions if DAQ board does not exist
+ statistics/	Statistics functions
+ time/	Time calculation functions
+ misc/	Miscellaneous library routine
+ spike_sorting/	Spike sorting assistant programs
+ kdf/ + perfmon/ + spike_collector_if/	Obsolete or personal use
test/	Miscellaneous test routines
modeling/, prj/ spike_collector/ template/	Obsolete or personal use
count_line.sh	Count line
endkm.m startkm.m	Add path and remove path

d. Coding convention

e. Troubleshooting
   i. Know where the bug occurs - H/W ? Interface ? Software ?
      1) Check H/W -> Software
      2) Use oscilloscope to determine source of the problem

f. Tour plan
   i. We will explore one program for one or two days.

Calibration and task execution

2008년 1월 10일 목요일

오후 7:00

1. Global structures

gScrInfo	screen and monitor calibration information
gEnv	Environment for DAQ object and graphic handles
gParam	Parameter for paradigm
gCalibData	Calibration data
TrialInfo	Trial information
TrialData	Trial data

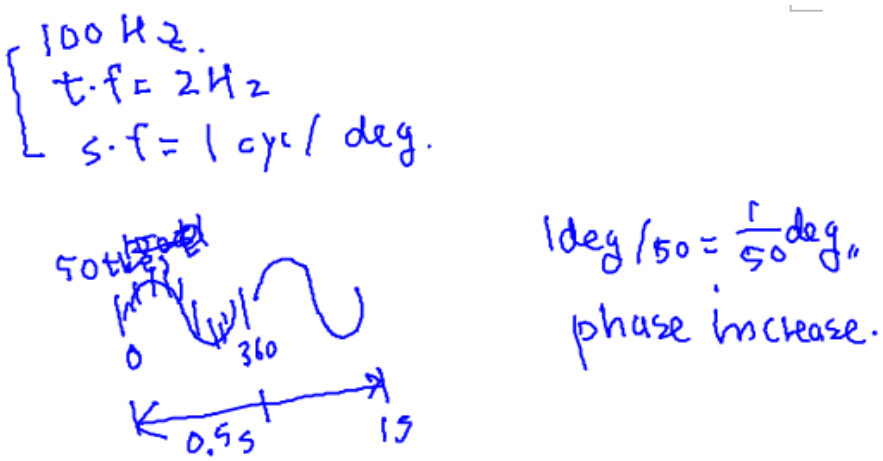
2. Read eye signal
    a. gEnv.nPeekData = 10 points average
    b. Analoginput structure
        i. Refer explanation in Workspace variable viewer
        ii. SamplePerTrigger: prepare memory for input data
        iii. SensorRange, InputRange, UnitsRange
3. CalibMainPannel
    a. Calibration method
        i. Linear
        ii. Cosine
        iii. Interpolation
    b. Data Structure
        i. Synchronize GUI edit box variables - global data structure
            1) UpdateUIFromParam.m, UpdateGainOffsetFromParam.m
            2) GetGainOffsetFromUI.m, GetParamFromUI.m
    c. Use case
        i. Open CalibMainPannel
        ii. Configure popup menu and variables
        iii. Push GO button
        iv. Push SAVE if needed
    d. Flow in GO
        i. UI -> global variables (gParam, gCalibData)
        ii. Switch gParam.strCalibMethod
                Case 'Paradigm name'
                    <Execute each paradigm functions>
        iii. End
    e. Example of paradigm functions: CalibFlinkSti.m
        i. Load image
        ii. Make stimulus
        iii. Calculate positions
        iv. Start loop
            1) Display stimulus
            2) Catch keyboard input
            3) Update global variables (gCalibData)
        v. End loop
        vi. Update UI (UpdateGainOffsetFromParam.m)
    f. How to debug
        i. Check tag name at Guide
        ii. Check global structure variable at UpdateXXX.m
        iii. Find functions with "Find Files.."
    g. Homework
        i. Slower pursuit speed
        ii. Use another stimulus images for fixation
        iii. Make RMSE-based offset calculation routine

Rotating Dimension

2008년 1월 12일 토요일

오전 10:00

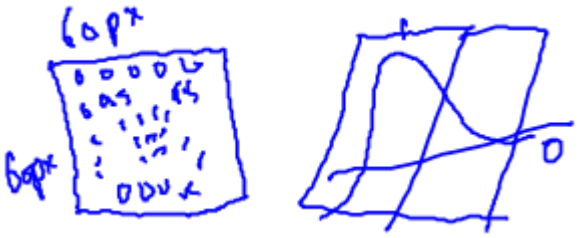
1. RotDim (RF_RotDimGr.m)
    a. Make Stimulus
        i. Smooth edge
        ii. Moving gratings
            1) All stimulus is made before displaying (PutImage)
            2) Only CopyWindow at runtime
            3) Each stimulus configuration has its own filename



화면 캡처: 2008-01-12, 오후 8:37

    a. Global Structure
        i. gRFInfo structure having receptive field parameters and ranges for rotation dimension
        ii. GInfo variables and stimulus handles for displaying stimulus (rect size, offscreen handle, etc)
    b. Main routine (RF_RotDimGr)
        i. Check start mode
            1) If View mode, send string command 'View' to RTMonitor
            2) If Save mode, send string command 'Save' to RTMonitor
        ii. Initialize gRFInfo structure to default stimulus
        iii. Set brightness array to distinguish parameter order
        iv. Start DAQ
        v. Iterate loop
        While (press 'q' quiton)
        Set the rectangular position to display stimulus
        CopyWindow stimulus handle to the rectangular position
        Check keyboard. If key is pressed,
            Change rotating parameter or parameter variables
            bRedraw = 1
            (If press 'q', stop iteration)
        If bRedraw == 1, make or load stimulus by GetGrating()
            If pre-loaded image exist, release them.
            Ginfo = GetGrating
        Check eye position and give reward
        vi. Stop DAQ

c. Get stimulus warpper function (GetGrating)
    i. If pre-saved stimulus file exist, load it and return GInfo
    ii. Otherwise, Call MakeRotateDimGrating_MW and save the stimulus(GInfo) into file
d. MakeRotateDimGrating_MW (hStiMaker is function handle)
    i. Calculate proper number of frame considering frame rate, temporal frequency
    ii. Get parameter array of rotating parameter
    iii. Iterate parameter array
        1) Make stimulus with given parameters (fixed parameter + one element in rotating parameter array)
        2) Apply gaussian filter (mask)



    i. Make stimulus handle with 'OpenOffScreenWindow' and 'PutImage'
    ii. Assign proper stimulus size
    iii. Return GInfo structure with necessary parameters

RFViewer, RFTrialMaker, RFStimulator

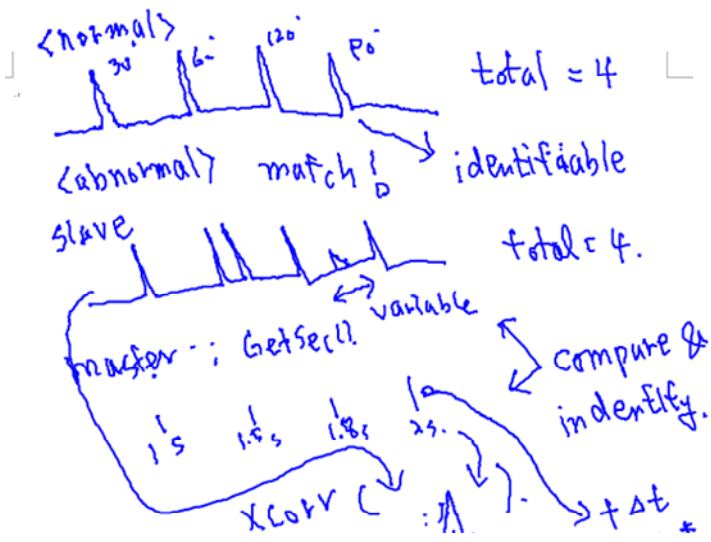
2008년 1월 16일 수요일

오후 7:00

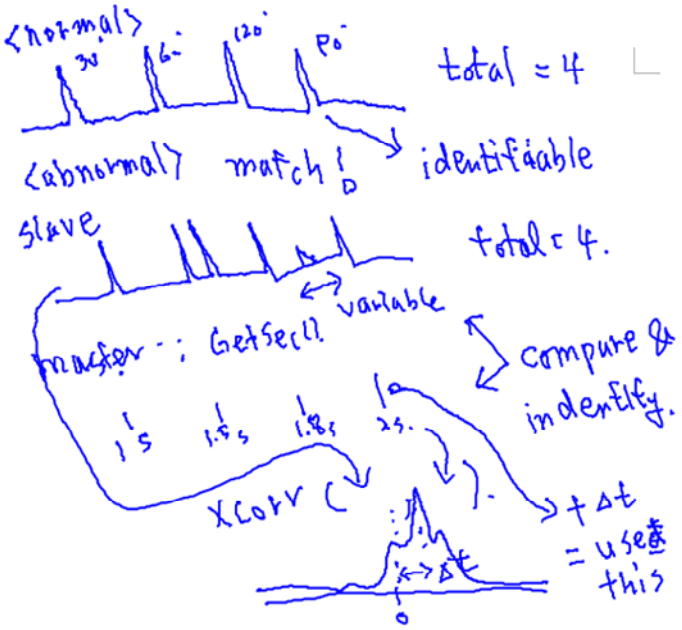
1. RFViewer
   a. UI interface to call RFTrialMaker, RFStimulator
   b. RFViewer_UpdateGRFInfoFromUI: UI values -> gRFInfo structure variables
2. RFTrialMaker
   a. Run mode along
      i. WGN
      ii. SubField
      iii. PSTH-based (forward correlation)
   b. Make stimulus
      i. Stimulus file with the same condition exists, load it. Otherwise, make it and save it to the file
      ii. Multi-dimensional parameters are ordered in one-dimensional value => Stimulus ID
         1) Stimulus ID is provided at each stimulus and used in STA method
      iii. Stimulus is saved to StiImage cell array (e.i. StiImage{1} = 30*30 int8 array
   c. Manipulate multi-dimension parameters
      i. CondListGenerator: make conditions with respect to parameter array
         1) H, V, diameter, orientation, phase, spatial frequency, contrast, luminance, temporal frequency
         2) Total condition number = ∏ number of parameters per condition
      ii. CondSeqGenerator: make trials based on trial sequence method.
         1) Total trial number = condition number * number of trials per condition
      iii. CondSpace, CondList, CondSeq, CondSize
      iv. Dimension of Interest (DOI)
   d. Global structure
      i. gRFInfo has parameters and RF mapping configuration. gRFInfo.cParam contains parameters in each dimension
      ii. TIPATH: most recently made TrialInfo file path
   e. Make TrialInfo (RFTrialMaker_Ext.m)
      i. Load stimulus if such a stimulus file exists
      ii. Otherwise, make stimulus. Iterate all stimulus parameters
         1) Make stimulus with respect to each parameters
      iii. End
      iv. Save stimulus into file
      v. Save TrialInfo file
      vi. Return TrialInfo path -> assign to TIPATH

3. RFStimulator (RFStimulator_Ext.m)
   a. Display stimulus when subject fixates
   b. Notice exact display time to RTMonitor by photodiode
      i. Save display time in seconds in case of inequality between photodiode and CondSeq number
      ii. Get cross-correlation between seconds data (sStiOnset) and photodiode data
      iii. Synchronize sStiOnset by adapting (add or subtract) argmax(xcorr(sStiOnset, Photodiode onset))

i.



i.



c. Main routine
   i. Save and transfer calibration file
   ii. Send TrialInfo file
   iii. Convert stimulus image to image handles
   iv. Send "START" command to RTMonitor
   v. Iterate total trials
      1) Check eye position
      2) Compensating eye movement if necessary
         a) Add gaze pixel to pxRect array
      3) WaitBlanking 1 frame
      4) Remove previous image
      5) Display current image
      6) WaitBlanking remaining frames
   vi. End
   vii. Send "End" command to RTMonitor
   viii. Send TrialInfo file

RFManager, RFAnalyzer

2008년 1월 18일 금요일

오후 7:00

1. RFManager (RFManager_Ext.m)
    a. Communicate with RFStimulator and save data to .daq file
    b. Main routine
        i. Receive TrialInfo file
        ii. Start DAQ
        iii. Iterate
            1) Check 'End' or 'Abort' command in network nonblock mode
                a) If command received, break loop
            2) Read data from DAQ and display data
        iv. End DAQ
        v. Receive TrialInfo file, stimulus image
2. RFAnalyzer (RFAnalyzer_Ext.m)
    a. Memory management
        i. Large memory usage
            1) 1channel, 2min data: 2min * 60sec/min * 25000samples/sec * 8 bytes / sample = 22MB
            2) Therefore, if we load eyeH,eyeV, photodiode, cell simultaneously, it would be 88MB.
            3) We need more memory to process these data: memory full error
        ii. Strategy to save memory
            1) Load each channel sequentially when the data is needed and release memory
            2) Current DAQ resolution is 16bit. So, load data with int16 (2 byte) data type instead of double (8 byte)
    b. Main routine
        i. Read TrialInfo
        ii. Load photodiode data and check stimulus count
            1) Get Photodiode stimulus onset time (GetPhysicalStiOnset.m)
            2) Use photodiode onset time if Master stimulus count == Slave photodiode count
            3) Synchronize Master stimulus time if above does not match (GetExactStiOnsetTime_ByXCorr.m)

Diode > Threshold

[Onset Offset] = GetFlipTime(photodiode > threshold)

        iii. Forward analysis
            1) Make Event(stimulus)-related spike array (MakeEventRelatedSpike_IT.m)
            Two variables with different temporal range
                a) cSTSViewData: data structure for rasterplot (larger range e.i. [-100 400])
                b) cSTSResData: data structure for getting response (e.i. [40 200])
            2) Make group result for event-related spike array (MakeGroupResult.m)
                a) Two data structure in each variable
                    i) SpikeSum: Accumulated data for rasterplot
                    ii) SpikeStream: Summed data for histogram, mean firing rate
            3) Plot PSTH (PlotSTS_Ext_Array.m)
            4) Fit result (FitGroupResult.m)
                a) For orientation, make circular duplicate (0:30:150 -> expand to -30:180) and fit

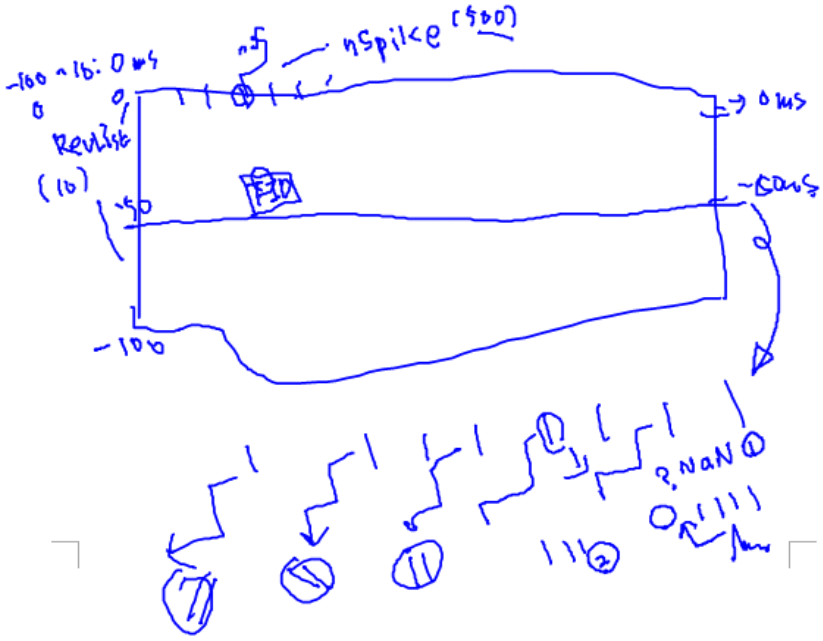
b) For other parameters, fit with linear DoG

5) Get appropriate index (GetRFIndex.m)

6) Save result

iv. Reverse analysis

1) Make Spike-triggered trial ID (GetSpikeRelatedTID.m)



2) Extract each stimulus parameter in DOI with respect to trial ID

3) Plot reverse correlation (Plot2DRevCorr.m)

4) Save result

v. STA analysis

1) Make spike-triggered stimulus ID (GetSpikeTriggeredStimulusHist.m)

2) Load stimulus

3) Shift image considering fixational eye movment (if necessary) (MakeShiftImage.m)

4) Make STA image array (MakeSTA.m)

5) Plot STA (PlotSTA.m)

6) Save result

RTMonitor

2008년 1월 22일 화요일

오후 7:01

1. RTMonitor
    a. Fast calculation for real-time spike filtering
        i. Use array calculation instead of explicit loop
        ii. Use hist() function in case of counting something
        iii. Lessen memory usage with logical data type
    b. Global data structures
        i. gRTM
            1) ai: DAQ object
            2) vUpDisc, ...: Spike filtering parameters
            3) bPlotXXX, ...: Analyze and plot selection
            4) WFProp : waveform shape (max-min / min-max)
        ii. gRTMData
            1) Channel data
                a) Data: working channel
                b) ReadData: Total channel
            2) Intermediate data for spike processing
    c. Main Routines
        i. FindSpike: identify spike with given parameters
            1) Load spike channel
            2) Differentiate data
            3) Get max-min or min-max time (with respect to waveform shape)
            4) Filter by given constraints
                a) Upper threshold
                b) Lower threshold
                c) Peak-to-peak interval
                d) Peak-to-peak height
        ii. AnalSpike: analyze activity (on-demand with checkbox)
        iii. PlotSpike: plot raw data or analyzed results
    d. Communication protocol in Network mode (RTM_IterCommand.m)

Appendix A: Citation about Channel Coupling problem

2008년 1월 10일 목요일
오후 11:55

Cited from NI homepage (source is noticed at the bottom)

Coupling between two AI channels

Hi there,

I am using PXI 6259 for acquiring data from two loadcells. I am using differential settings and have to use only one and two loadcells depending on the situation. When both loadcells are connected everything works fine but when I disconnect the second load cell from the DAQ card then and start experimenting with only one load cell i find that both the AI channel which is not connected anymore is following the other AI channel in use. Also in this situation there seems to be a lot more noise in the readings. I would expect the channel not in use to saturate etc. but seeing it follow the other channel exactly is hard to comprehend. Is there an explanation for this situation.

Thanks and Regards,
Viku

Re: Coupling between two AI channels

It is normal and defined as crosstalk:
http://zone.ni.com/devzone/cda/tut/p/id/630
With the 2nd channel disconnected are you saying the the readings on the connected channel are noisier?

~~~~~~~~~~~~~~~~~~~~~~~~~~
'It's the questions that drive us."
~~~~~~~~~~~~~~~~~~~~~~~~~~

Re: Coupling between two AI channels

The 6259 uses CMOS multiplexers to switch between input channels; these multiplexers are more or less connected directly to the input terminals of the board. They dominate the specified 100 pF input capacitance of the board, and hold a lot of charge as a result.

On the other hand, the input resistance is very high (> 10 Gohms), and so any charge left on an otherwise unconnected line will determine the voltage on the line. When the multiplexer switches from your connected load cell channel to

the unconnected (floating) channel, the charge stored from the first channel remains in the multiplexer circuitry and holds the voltage where it was until a lower impedance is connected and can influence it. This is why the unconnected channel appears to follow the connected one.

I suppose if the board is allow to sit on the unconnected channel for several seconds without switching back and forth the input will saturate, but if it doesn't, it's testimony to the very high input resistance of the board.

I'm not sure I can explain why you're seeing more noise in that situation, though. More information would be needed. What kind of noise? Can you supply plots?

EBL

Re: Coupling between two AI channels

Thankyou very much for your reply. Yes now I understand the why the spurious signal is happening at the disconnected channel (basically the impedance is infinite and does not allow the capacitor to discharge at all and hence.....)

For the second part, when the channel is disconnected I start getting more noise in the channel that is connecting. Also the noise seems to depend a lot on the location and environment of the unconnected connector hanging out of the breakout box. For example as soon as I touch it I start seeing a 55Hz noise amplitude in the other connected signal. I was trying to relate it with AC mains noise but could not explain it to myself. Any suggestion would be highly appreciated.

regards
viku

Re: Coupling between two AI channels

Well, it certainly sounds like crosstalk to me. On this board (and most other boards with multiplexed front-ends) there are several ways for crosstalk to occur:

1) Capacitive coupling: signals are coupled by capacitance between adjacent wires, either in the cabling or on the board itself. (This could happen on any board - not just ones with multiplexers). Since it's capacitive, it's frequency-dependent, and high frequencies couple much better than low frequencies. Since you're seeing ~55 Hz, this is probably not the mechanism behind your problem.

2) Channel-to-channel settling crosstalk: signals on one channel influence the measurement on the next channel in the scan sequence. When the multiplexer switches from one signal to the next, the signal at the A/D converter does not switch instantaneously from one to the other. Due to limited bandwidth of the amplifier and the A/D converter, and capacitance in the multiplexers, there is a settling time as the signal transitions from one level to the next. At first blush it usually has the shape of an exponential function, though once that part has settled out there's usually slower effects that aren't exponential. Those slower components of the settling curve are often due to thermal transients and dielectric absorption in capacitors. In any case, if the A/D converter samples the signal before it's through settling adequately to the new level, the new signal will carry some residual influence of the previous channel. Since the curve usually decreases monotonically, waiting longer to settle decreases the amount of influence. So one thing to try is to lower the channel-to-channel sample rate within each scan - if that decreases the noise, then your problem is likely related to settling.

3) Track-and-hold feedthrough: this is not as common as the previous effect, but it can happen in some systems. While the A/D converter is busy converting a signal that's being held steady by its track-and-hold, oftentimes the multiplexer switches to the next channel to allow the amplifier and other circuitry time to settle before the next sample is taken. The resulting transient can couple a little of the signal from one channel into the measurement of the -previous- channel since the A/D converter is still working away on the previous channel. One way to test for this effect would be to tie the same signal to two (consecutive) channels instead of one, and scan three channels instead of two. Since the voltage at the A/D converter would be remain the same when the multiplexer switches to the second copy of the signal, there should be no feedthrough on the first conversion of the signal (but there still might be settling from the channel before that). I don't think this is the problem in your case, but it's worth investigating.

Hope this helps,
EBL

Re: Coupling between two AI channels

Thank you very much for the detailed reply EBL. Now I understand crosstalk. a lot better. And I feel that what I was seeing was definitely a cross talk problem. I was using coaxial cables with bnc connectors to connect the AI signals to SCH 68. And I guess the unconnected bnc connector was acting like an antenna picking up the 50Hz signal from a nearby CRT and was getting fed to the connected channel via crosstalk. Now the problem is solved by terminating the unconnected bnc connector with a proper terminator when not using it. Also the CRT is moved further away. So finally I am getting only a tolerable amount

of noise in my signal.

Besides all the replies that I got here from the wonderful people out there following documents on ni.com have also helped me in resolving this problem:

http://digital.ni.com/public.nsf/allkb/CE719E5A9409ADCC8625714100033433

http://digital.ni.com/public.nsf/allkb/73CB0FB296814E2286256FFD00028DDF

Once again thank you all very much.

Regards,

원본 위치 <http://forums.ni.com/ni/board/message?board.id=250&message.id=32822>

Appendix B: Current Platform Specification

2008년 1월 24일 목요일
오후 5:37

Current platform (2008/01)

- Recording room
  - Stimulator : - BNC-2110
    - PCI-6013 : Low-Cost 200 kS/s, 16-Bit, 16-Analog-Input Multifunction DAQ
    - h/p http://sine.ni.com/nips/cds/view/p/lang/en/nid/11441 manual http://www.ni.com/pdf/manuals/370636a.pdf
    - BNC-2110: Shielded Connector Block for M Series, E Series, and NI 67xx Devices
    - h/p http://sine.ni.com/nips/cds/view/p/lang/en/nid/1865 manual http://www.ni.com/pdf/manuals/321860e.pdf
  - Spike Collector :
    - PCI-6052E: 333 kS/s, 16-Bit, 16-Analog-Input Multifunction DAQ http://sine.ni.com/nips/cds/view/p/lang/en/nid/2601 http://www.nacse.org/neesSiteSpecs/incoming/32/equipmentDoc/403/403/6052E_Manual.pdf
    - BNC--2090: 22 BNC connectors for analog, digital, and timing signals / 28 spring terminals for digital/timing signals / Shielded, rack-mountable BNC adapter chassis http://sine.ni.com/nips/cds/view/p/lang/en/nid/1177

- M411
  - Right : PCI-6023E : 200 kS/s, 12-Bit, 16 Analog Input Multifunction DAQ
    - http://sine.ni.com/nips/cds/view/p/lang/en/nid/10967
    - BNC-2110 : Shielded Connector Block for M Series, E Series, and NI 67xx Devices http://sine.ni.com/nips/cds/view/p/lang/en/nid/1865
  - left(DAQ)
    - NuDAQ PCI-6208
    - not NI, no BNC board (directly connected to right) / 8/16 Channels Analog Output Cards
    - http://www.aberdeeninc.com/abcatg/DAQ100.htm http://www.aberdeeninc.com/abcatg/DAQ103.htm
- M414
  - CATS : PCI-6013 - BNC-2110